CHAPITRE 0

MODE D'EMPLOI

0.1 Organisation de cet ouvrage

0.1.1 Première partie: le langage (parler VHDL)

Il y a toujours deux façons d'aborder un langage: en bloc ou par analogie.

S'il aborde VHDL en bloc, le lecteur apprendra globalement toutes ses caractéristiques et fonctionnalités avant de réellement commencer à modéliser.

Le document décrivant la norme VHDL, aussi appelé LRM[1], répond sur le fond à une telle attente mais, sur la forme, il est parfaitement indigeste et impropre à cet usage. Il doit donc être considéré comme un guide pour les constructeurs d'outils basés sur ce langage ou comme une référence pour le concepteur confirmé confronté à un point obscur. Un lecteur non initié s'y perd rapidement. Ce n'est absolument pas un manuel utilisateur, et d'ailleurs, cela n'y prétend pas.

Une des entrées de ce livre (la première) est donc de présenter le langage globalement mais de la manière la moins ardue possible. Beaucoup de schémas et d'exemples agrémentent cette partie. Si elle commence sans originalité par un chapitre « Introduction », le chapitre « Pièges » qui la clôt mérite un commentaire.

Il offre une distrayante révision de ce qui a été exposé précédemment. Il consiste en une série, disons... d'exercices comportant chacun un énoncé suivi d'une description VHDL qui peut sembler (de prime abord) résoudre le problème. En fait, cette description ne « fonctionne » pas et les raisons de ce dysfonctionnement ainsi qu'une solution réelle à l'exercice sont données ensuite. Le lecteur pourra aborder ce chapitre comme une sorte de jeu (où est l'erreur ?) ou comme une introduction à la modélisation. Un rappel des principaux traits du langage est implicite.

Arrivé au terme de la lecture des caractéristiques du langage, le lecteur pourra considérer la seconde partie (la modélisation en VHDL) comme un vaste recueil d'exercices et d'exemples.

---

[1] « Language Reference Manual » qui signifie manuel de référence du langage pour les Anglais, qui comme chacun sait, écrivent de droite à gauche.

Un index et un aide-mémoire, localisés à la fin de l'ouvrage, l'aideront à venir à bout de ses premières modélisations. Par la suite, le glossaire de la fin lui permettra de trouver la correspondance des termes français utilisés ici par rapport à ceux évidemment anglais de la norme. C'est le prélude à son déchiffrage du LRM.

Cette démarche consistant à attaquer le langage de front devrait être celle d'informaticiens ou de personnes ayant de bonnes connaissances dans un autre langage de simulation fonctionnelle.

0.1.2 Deuxième partie: la modélisation (écrire VHDL)

L'autre approche, beaucoup plus pragmatique, correspond à la deuxième partie de cet ouvrage. Le lecteur choisira un problème, si possible voisin de ses préoccupations professionnelles, et regardera comment il est possible de le traiter en VHDL. Il pourra alors par analogie (et au moyen de références à la première partie par le biais de l'index) construire et tester une description VHDL traitant d'un problème légèrement différent. De proche en proche, il pourra ainsi continuer l'exploration avant de s'intéresser à un autre cas de figure.

Dans cette partie du livre sont présentés des exemples de modélisations dans des domaines variés. A chaque fois, le problème est posé et une ou plusieurs solutions sont données.

En tête de chaque chapitre sont nommées les plus importantes notions du langage qui y sont illustrées. La recherche de cette notion dans la première partie (via l'index) permettra au lecteur de parfaire progressivement ses connaissances.

Chaque chapitre se finit par une ou plusieurs propositions d'exercices.

0.1.3 Troisième partie: la synthèse (matérialiser VHDL)

La grande majeure partie des utilisateurs des langages de description matérielle, et de VHDL en particulier, vise la réalisation matérielle. Il est même sensé de dire que le développement de VHDL repose presque en totalité sur la synthèse. La pratique des techniques de synthèse est devenue un passage obligé pour la conception des systèmes matériels, qu'ils soient ASICs[1] ou circuits programmables de type FPGA ou PLD[2]. Les techniques de synthèse elles-mêmes sont en grande mutation (l'évolution vers la synthèse architecturale est abordée).

Cette troisième partie s'attache à expliciter le plus simplement possible la sémantique du langage VHDL en vue de la synthèse logique (rappelons que le langage VHDL a été spécifié pour la simulation).

Dans cette partie du livre sont décrits les dispositifs logiques fondamentaux. Il sera aussi aborder la description de systèmes plus complexes tels que les machines d'états, les mémoires...

---

[1] ASIC pour *Assembled Specific Integreted Circuit*, donc des circuits à la demande composés d'un assemblage de portes logiques élémentaires et, éventuellement, de blocs durs tels que des mémoires ou encore des cœurs de processeurs.

[2] FPGA désigne : *Field Programmable Gate Array* et PLD : *Programmable Logic Device*. Ces deux types de composants sont en train d'envahir le marché...